

Contents

1 Routine/Function Prologues	2
1.0.1 time_interp_gdas (Source File: time_interp_gdas.F90)	2
1.1 Core Functions of time_interp_gdas	2

1 Routine/Function Prologues

1.0.1 time_interp_gdas (Source File: time_interp_gdas.F90)

Opens, reads, and interpolates GDAS forcing.

TIME1 = most recent past data

TIME2 = nearest future data

The strategy for missing data is to go backwards up to 10 days to get forcing at the same time of day.

1.1 Core Functions of time_interp_gdas

zterp Performs zenith angle-based temporal interpolation

REVISION HISTORY:

```

1 Oct 1999: Jared Entin; Initial code
25 Oct 1999: Jared Entin; Significant F90 Revision
11 Apr 2000: Brian Cosgrove; Fixed name construction error
              in Subroutine ETA6HFILE
27 Apr 2000: Brian Cosgrove; Added correction for use of old shortwave
              data with opposite sign convention from recent shortwave data.
              Added capability to use time averaged shortwave & longwave data
              Altered times which are passed into ZTERP--used to be GMT1
              and GMT2, now they are LDAS%ETATIME1 and LDAS%ETATIME2
30 Nov 2000: Jon Radakovich; Initial code based on geteta.f
17 Apr 2001: Jon Gottschalck; A few changes to allow model init.
13 Aug 2001: Urszula Jambor; Introduced missing data replacement.
5 Nov 2001: Urszula Jambor; Reset tiny negative SW values to zero.

```

INTERFACE:

```
subroutine time_interp_gdas()
```

USES:

```

use lisdrv_module, only :lis, grid
use baseforcing_module, only: glbdata1, glbdata2
use time_manager
use time_module, only : tick, time2date
use grid_spmdMod
use spmdMod
use gdasdomain_module, only : gdasdrv

```

CONTENTS:

```

if(masterproc) then
  if (get_nstep() .eq. 0) then
    lis%f%force = gdasdrv%nmif
  else

```

```

    lis%f%nforce = lis%f%nf
  endif
  endif
#endif(SPMD)
  call MPI_BCAST(gdasdrv%gdastime1,1,MPI_REAL8,0, &
    MPI_COMM_WORLD,ierr)
  call MPI_BCAST(gdasdrv%gdastime2,1,MPI_REAL8,0, &
    MPI_COMM_WORLD,ierr)
  call MPI_BCAST(lis%t%time,1,MPI_REAL8,0, &
    MPI_COMM_WORLD,ierr)
  call MPI_BCAST(lis%t%gmt,1,MPI_REAL,0, &
    MPI_COMM_WORLD,ierr)
  call MPI_BCAST(lis%f%nforce,1,MPI_INTEGER,0, &
    MPI_COMM_WORLD,ierr)
  call MPI_BCAST(lis%f%FOO_flag,1,MPI_INTEGER,0, &
    MPI_COMM_WORLD,ierr)
  call MPI_BCAST(lis%f%F06_flag,1,MPI_INTEGER,0, &
    MPI_COMM_WORLD,ierr)

#endif
  btime=gdasdrv%gdastime1
  call time2date(btime,bdoy,gmt1,byr,bmo,bda,bhr,bmn)
  btime=gdasdrv%gdastime2
  call time2date(btime,bdoy,gmt2,byr,bmo,bda,bhr,bmn)
  if ( (lis%f%F06_flag==0) .and. (lis%f%FOO_flag==1) ) then
    inittime = gdasdrv%gdastime2
    call time2date( inittime, idoy, igmt, iyr, imo, ida, ihr, imn )
    its = -6*60*60
    call tick( inittime, idoy, igmt,iyr,imo,ida,ihr,imn,iss,its)
  end if
  wt1 = (gdasdrv%gdastime2-lis%t%time) / &
    (gdasdrv%gdastime2-gdasdrv%gdastime1)
  wt2 = 1.0 - wt1
  do f=1,lis%f%nforce
    if ( f == 3 ) then !shortwave
      do c = 1, gdi(iam)
        zdoy = lis%t%doy
        if ( (lis%f%F06_flag==0) .and. (lis%f%FOO_flag==1) ) then
          call zterp( 0, grid(c)%lat, grid(c)%lon, igmt, gmt2, &
            lis%t%gmt,zdoy,zw1,zw2,czb,cze,czm,lis)
        else
          call zterp( 0, grid(c)%lat, grid(c)%lon, gmt1, gmt2, &
            lis%t%gmt,zdoy,zw1,zw2,czb,cze,czm,lis)
        end if
        grid(c)%forcing(f) = zw1 * glbdata2(f,c)
        if (grid(c)%forcing(f) < 0) then
          print *, '2 warning!!! SW radiation is negative!!'
          print *, 'sw=', grid(c)%forcing(f), '... negative'
          print *, 'gdas2=', glbdata2(f,c)
        end if
      end do
    end if
  end do
end program time_interp_gdas

```

```
        call endrun
end if

if (grid(c)%forcing(f).gt.1367) then
    grid(c)%forcing(f)=glbdata2(f,c)
endif
end do

else if ( (f==4) .or. (f==10) ) then
do c = 1, gdi(iam)
    if ( lis%f%F00_flag==1 ) then
        grid(c)%forcing(f)=2*glbdata2(f,c) -glbdata1(f,c)
    else
        grid(c)%forcing(f) = glbdata2(f,c)
    end if
end do

else if ( (f==8) .or. (f==9) ) then
do c = 1, gdi(iam)
    if ( lis%f%F00_flag == 1) then
        if (2*glbdata2(f,c) >= glbdata1(f,c)) then
            grid(c)%forcing(f)=2*glbdata2(f,c)-glbdata1(f,c)
        else
            grid(c)%forcing(f) = 0.0
        end if
    else
        grid(c)%forcing(f) = glbdata2(f,c)
    end if
end do
else
do c = 1, gdi(iam)
    grid(c)%forcing(f) = wt1 * glbdata1(f,c) +  &
    wt2 *glbdata2(f,c)
end do
end if
end do
return
```